Enabling the Migration Away from the ISA Bus

David Poisner
Senior Component Architect
Platform Components Division
Intel Corporation

Agenda

- Migration Challenges
- New Migration Enablers
  - Low PinCount I/F
  - AC'97
  - SMBus
- Migration Waves and Examples

Benefits of Removing ISA

- Reduces Platform Cost
  - Physical Implementation
  - Support for non-PnP
  - Simpler BIOS
- Lower support and fewer returns

PC’98 Recommends and
PC’99 Requires:
No ISA expansion

Other Improvements

- Eliminates undesirable system-level design
  - Asynchronous bus
  - Address limitations
- Latency and determinism
  - No unpredictable DMA slaves or ISA bus masters

Migration Challenges

- Difficult for some functions/features
  - Software infrastructure not ready
  - Institutional contracts may require specific hardware implementations
    - Ex: 2 serial, 1 parallel port, 1 FDC
- PCI not intended nor optimized for all migrations
  - External devices
  - Lots of simple slaves

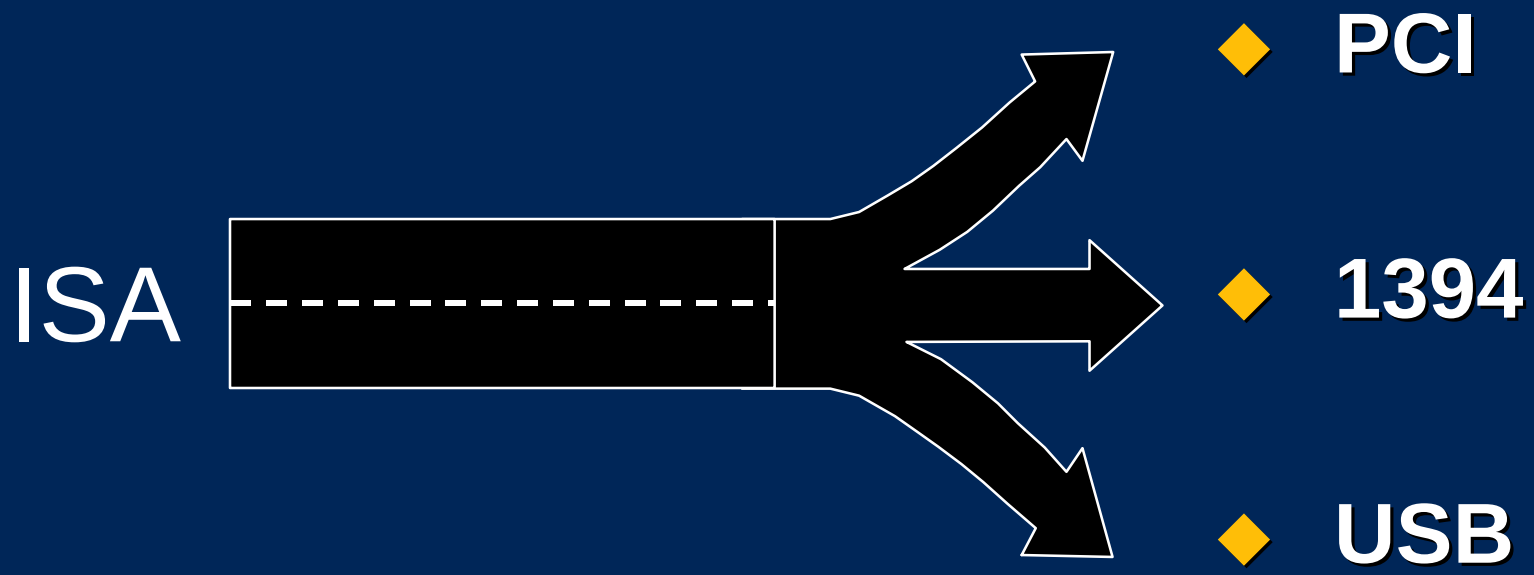
First … The Good News

- New platform functions and initiatives enable migration
  - Simple and quick hardware redesign
  - Minimize and streamline software changes
  - Lower H/W cost

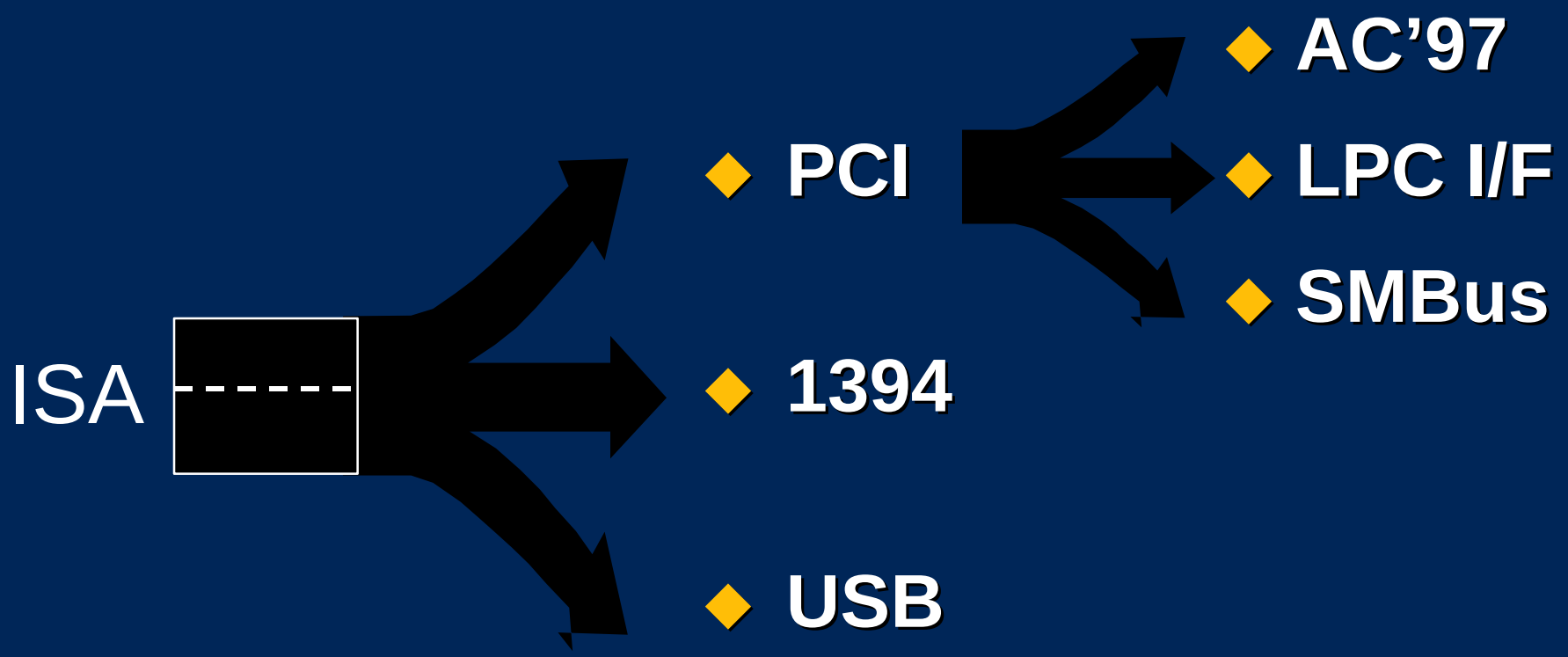
Also the Bad News...

- Choices, Choices, Choices

Existing Migration Paths



Clarifying the Paths



Options for Migration

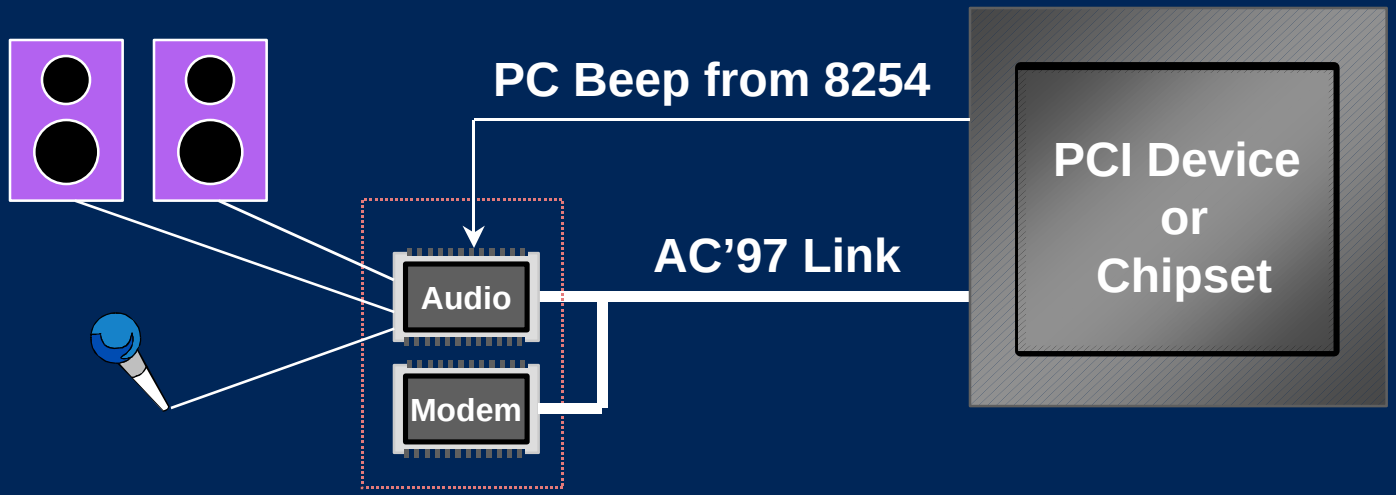
- PCI — Internal High-Speed I/O
- 1394 — High-Speed I/O and Storage
- USB — Low, Medium-Speed I/O
- AC’97 — Audio and Modem
- LPC I/F — ISA legacy functions, low-cost ASICs for internal functions
- SMBus — Very low-speed I/O

AC'97

- Rev 2.0 Specification Released
- Implemented as separate PCI devices
  - Audio, Modem
- SoundBlaster™ compatibility not provided by H/W
  - Handled at OS level
- Supports legacy PC "Beep"
  - Option to save internal speaker in some systems

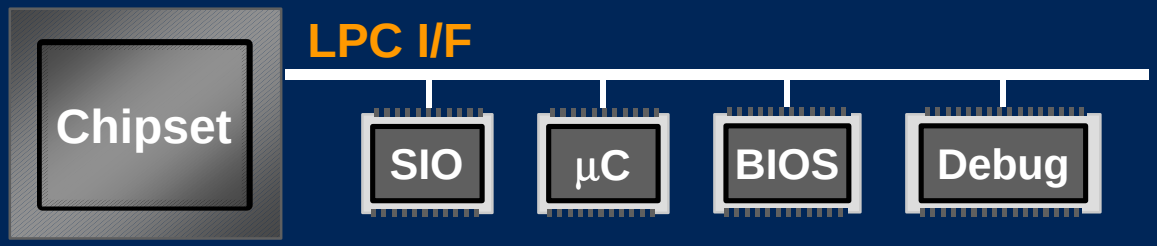
AC'97 Usage

- Codec separated from host controller
  - Higher fidelity
  - Modularity
- AC'97 Link supports multiple isochronous streams



LPC I/F Overview

- Low PinCount I/F
- For existing ISA and X-bus motherboard functions
  - Super I/O
  - Microcontrollers
  - Non-Volatile Storage
  - Simple prototype and debug logic



LPC is NOT an end-user expansion bus

LPC I/F Cycle Types

- I/O to traditional 64K space
- DMA for channels 0-7
- Memory space >> ISA or X-bus
- Performance ≥ ISA for all cycle types
- Bus Master cycles to memory and I/O space
  - No usage of 8237
  - Intended for microcontrollers

LPC I/F Signals

- Required Signals
  - LFRAME# Indicates start of cycle
  - LAD[3:0] Muxed address, data, control
  - LRESET# Reset (same as PCI Reset)
  - LCLK Same as PCI Clock
- Optional Signals
  - LDRQ# DMA/Bus Master request
  - LIRQ# Serialized IRQ
  - LPCPD# Powerdown indicator

LPC I/F Pin Requirements

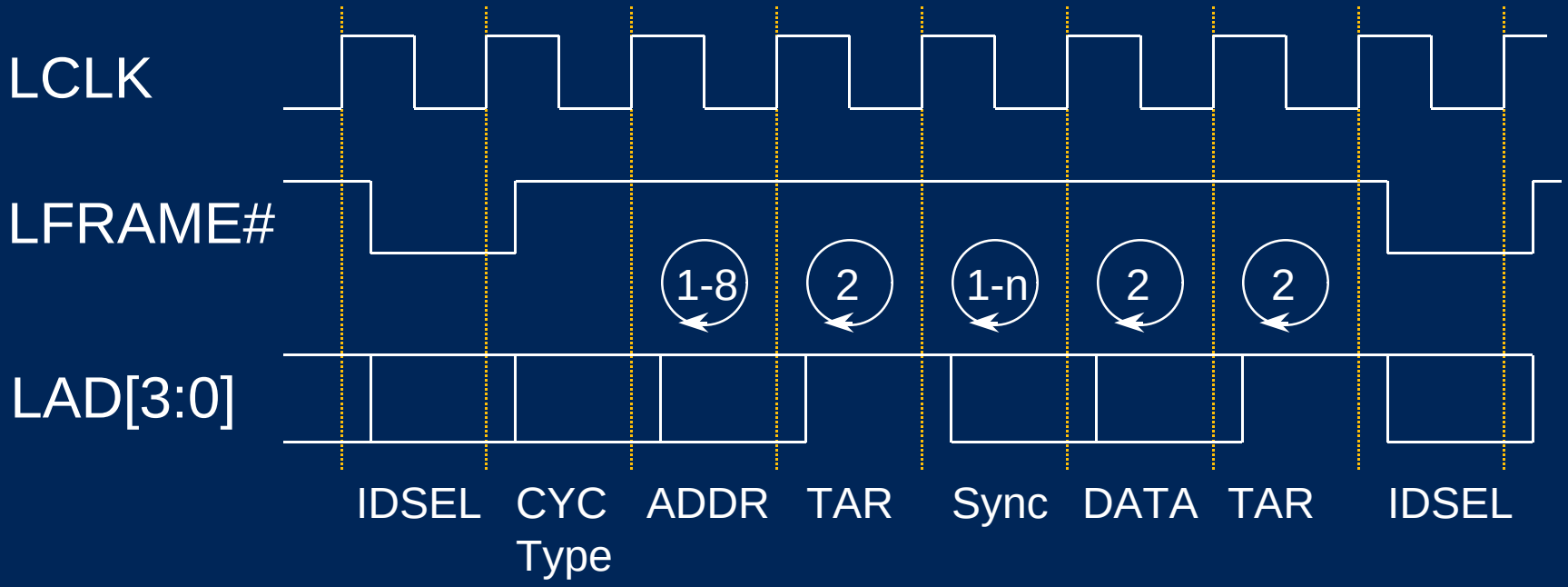
	ISA/X-bus Peripheral	LPC I/F Periph.
ISA Memory Target, 8-bit	34	7
ISA I/O Target, PnP		
8-bit / 16-bit	29/40	7
ISA DMA & I/O Slave, PnP	48	8
X-bus I/O Target	14	7
X-bus DMA & I/O Target	16	8

LPC I/F also saves 6-20 pull-up resistors!

Doesn't include interrupt signals
Only 1 interrupt for LPC I/F, many for ISA

LPC I/F Basic Cycle

- Multiplexed Command, Address, Data
  - Number of address, data clocks varies by cycle type



LPC I/F Robustness vs. ISA

	LPC I/F	ISA
Error Detection	Missing Target Too Many Wait States DMA Error	None
Error Reporting	NMI#, SCI, SMI# (Cycle based)	IOCHK# (NMI#)
Error Recovery	Abort Cycle	Full Reset

Other LPC I/F Capabilities

- LPCPD# (powerdown) signal
  - Allows isolation during Sleep states
- PCI CLKRUN# for mobile
- Protocol allows overlay and muxing of other cycle types
- Electrical specs same as 3.3V PCI
  - Allows for many devices and reasonable layout

LPC I/F Limitations

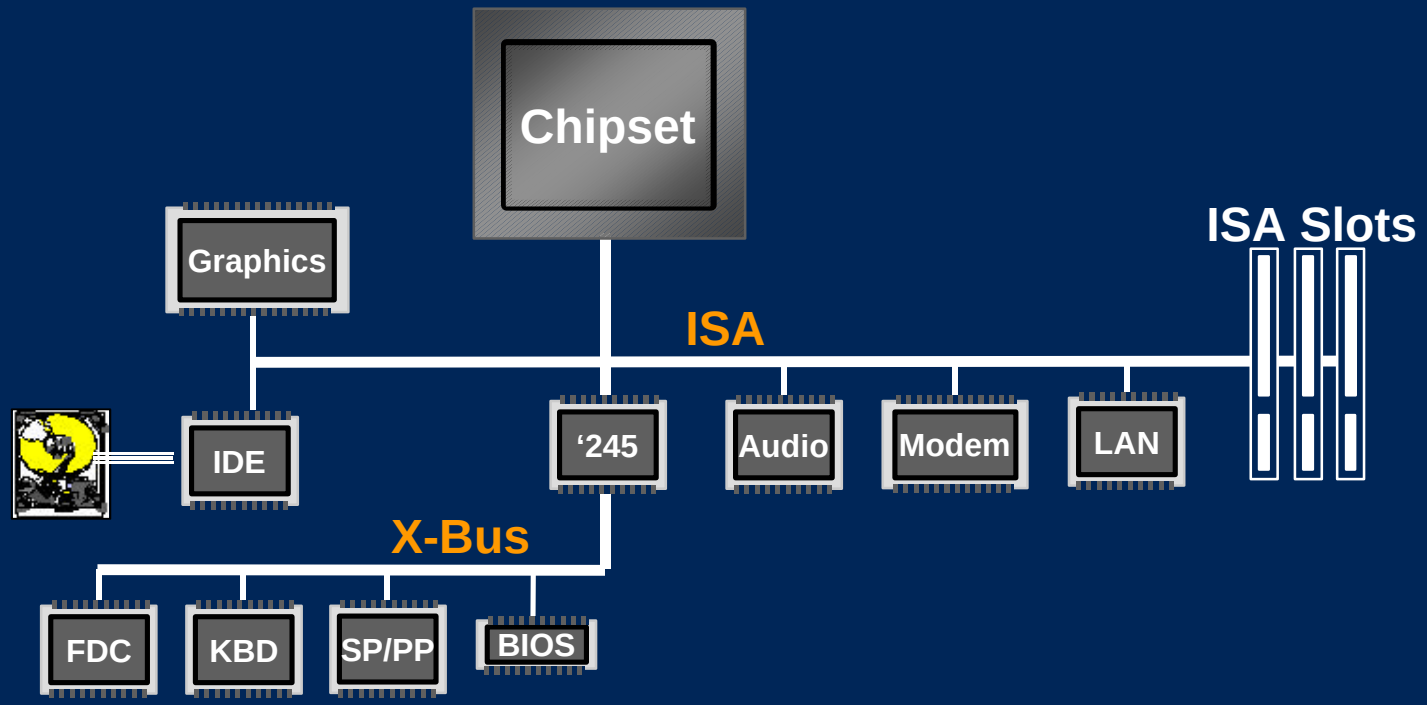
- DMA
  - No 8237 cascade mode
    - No ISA masters
  - Must be able to be preempted
    - Like Type-F DMA
- No peer-to-peer
- Bus master bursts limited to 32-bits
- All configuration handled via ACPI
  - No ISA PnP compatibility

SMBus

- Intended for serial interface
- Less than 100 kbps
- Well suited to low-cost functions
  - Temperature and voltage sensors
  - GPIO expanders
- Host Controller placement options
  - As PCI function in chipset
  - In embedded microcontroller

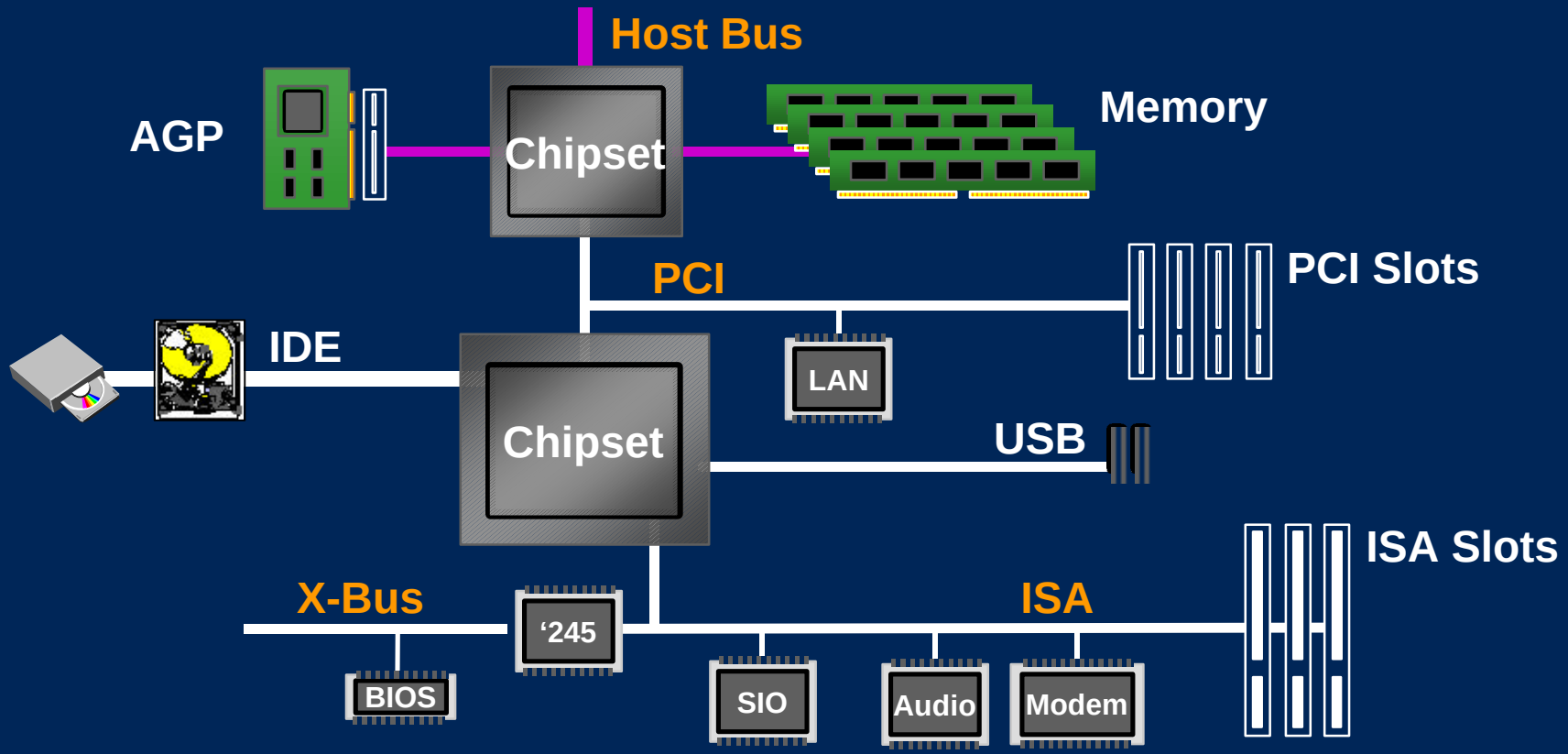
Migration Waves

- Prior to first wave, everything on ISA or X-bus



First Migration Wave

- Graphics: ISA-> PCI -> AGP
- LAN, IDE, USB: ISA -> PCI or Chipset

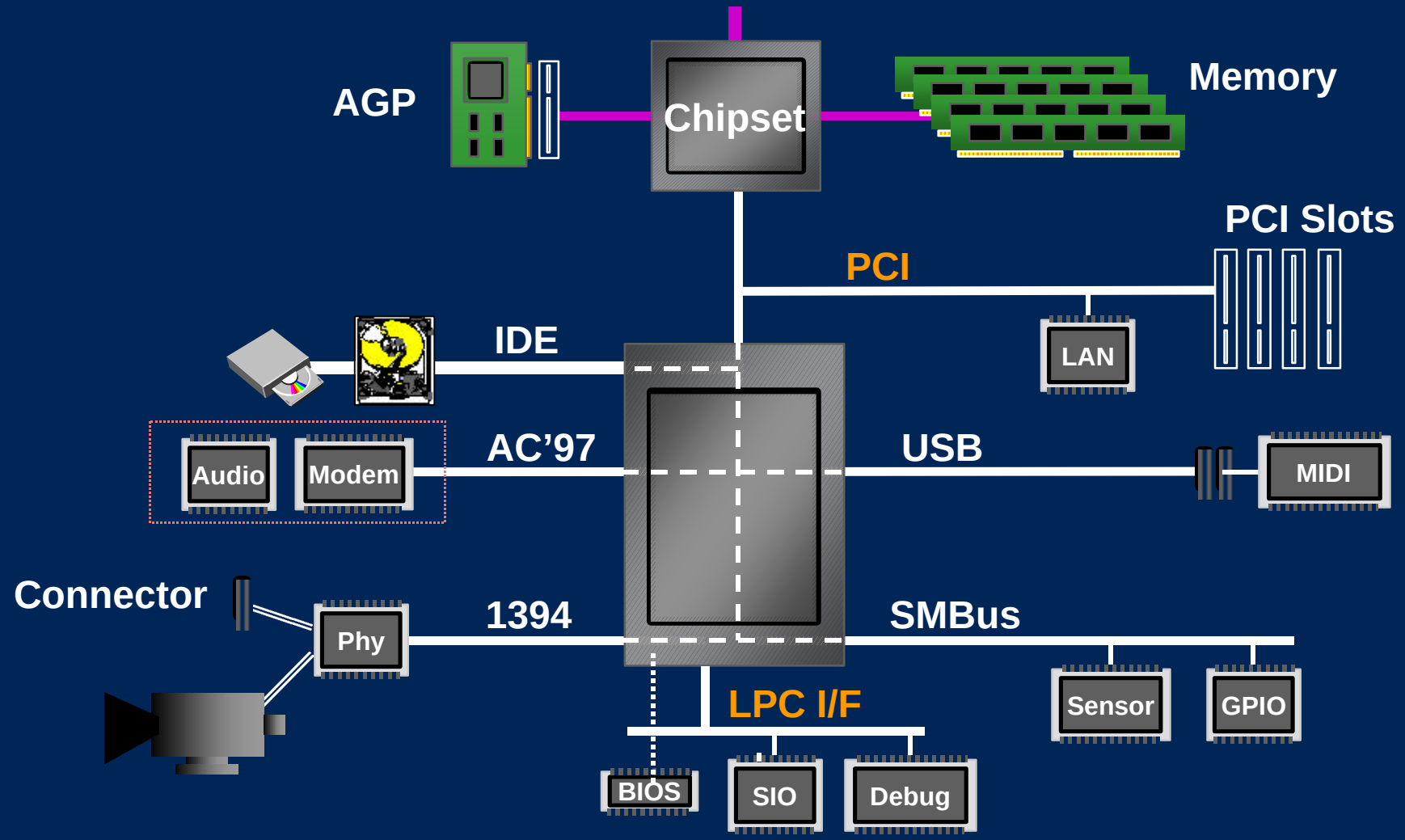


Next Migration Wave

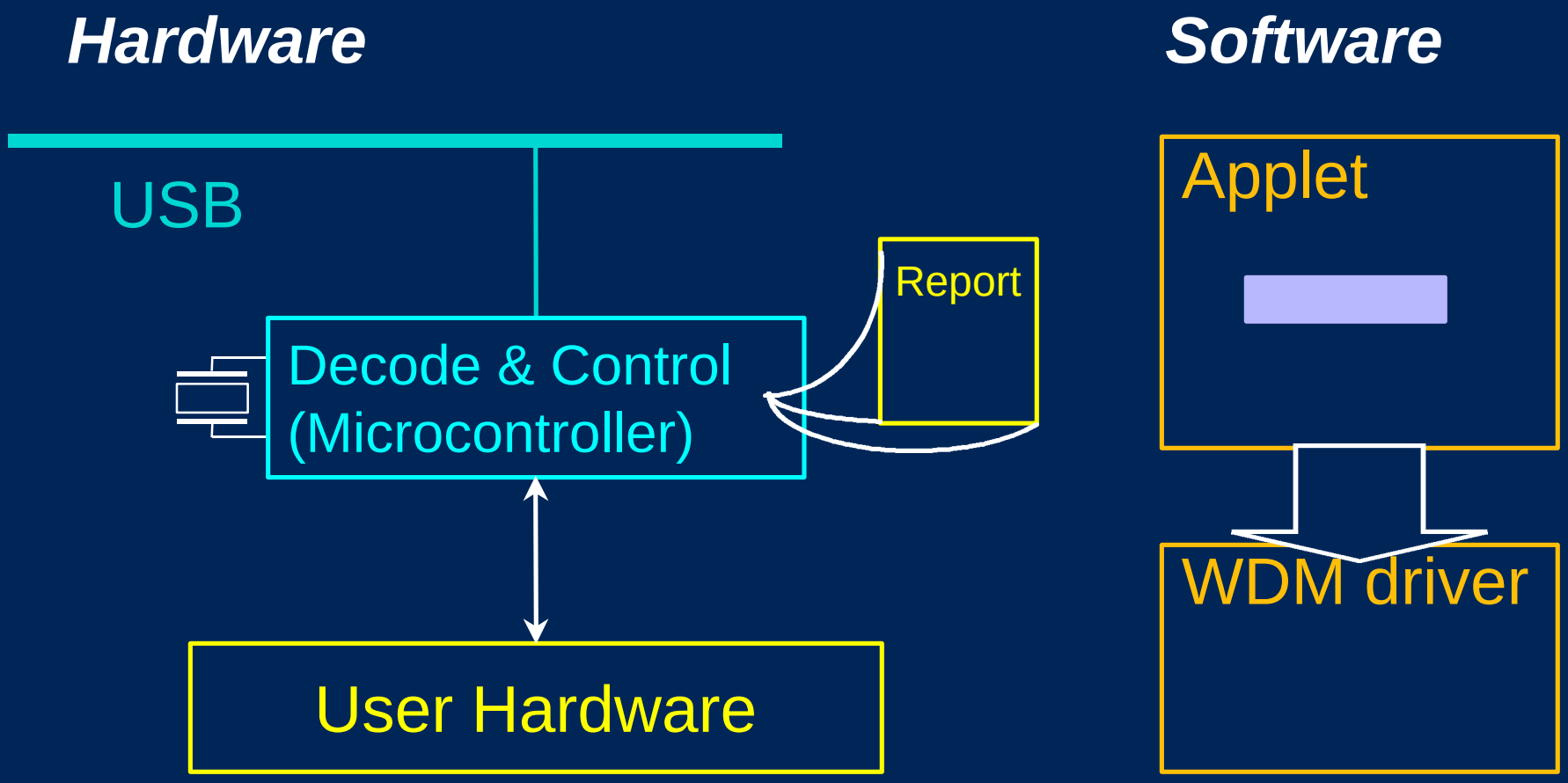
Function	Migrates To:
Super I/O	LPC I/F, USB, PCI
Audio	AC'97, USB, 1394, LPC*
Non-volatile Storage	LPC I/F or private I/F
MIDI UART	USB, LPC I/F
Legacy Gameport	USB, LPC I/F, or gone
µC	LPC I/F
Debug or Prototype	LPC I/F

- Legacy compatible audio possible on LPC I/F, but not encouraged

Example of Next Wave



USB Peripheral Migration
Design Example



Configuration complexity hidden by µC

USB Peripheral Migration
Design Tasks

- Hardware
  - USB I/F via μC
  - Specific functions via Logic + Firmware
- Firmware
  - Device information
- Software
  - Specific code within WDM template

USB Peripheral Migration
Report Descriptor

- Report is table driven, flexible
  - Can have multiple models with 1 software
  - µC does logical to physical device mapping
- Expanding Report Descriptors
  - Allows for more features and functions

USB Peripheral Migration
Other Implications

- BIOS/Chipset level hooks
  - Pre-boot support
  - Non-USB aware OS's
- OS-level support
  - Demonstrated here at WinHEC 98

Recognize Long-Term Trends

- Human I/F and external I/O via USB, 1394
- High-speed new functions via PCI
- LPC I/F continues for low-speed internal devices

Next Steps (Call to Action)

- Continue migration away from ISA
- Design using LPC I/F, AC’97, SMBus
- USB and 1394 for External PnP

Links to Specs

- LPC I/F
  - http://developer.intel.com/design/pcisets/lpc/
- AC'97 Rev 2.0
  - http://developer.intel.com/pc-supp/platform/ac97/
- SMBus
  - http://www.sbs-forum.org/specs.htm